Modelagem de Classificação para Determinação de Sucesso em Aplicações de Subsídios

1st Gabriel Vasconcelos Fruet
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
gabrielfruet@alu.ufc.br

2nd Kelvin Leandro Martins
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
kelvinleandro@alu.ufc.br

3rd Mateus Pereira Santos
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
mateussantos14@alu.ufc.br

4th Pedro Leinos Falcão Cunha
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará
Fortaleza, Ceará
pedrofalcao@alu.ufc.br

***Abstract*—Nesse trabalho são explorados diferentes modelos de classificação para prever o sucesso de aplicações para financiamento de pesquisas, a partir de dados da Universidade de Melbourne. Técnicas como regressão logística, *K-Nearest Neighbors* (KNN) e Redes Neurais foram utilizadas. Os modelos foram avaliados utilizando métricas como a acurácia, a matriz de confusão e a curva ROC. Os resultados determinaram que, enquanto os modelos não lineares conseguem capturar dinâmicas mais complexas dos dados, para esse conjunto de dados o modelo não linear, apesar de sua simplicidade, possuiu uma acurácia maior, mostrando que nem sempre a solução mais complexa é a melhor.**

***Index Terms*—Predição de sucesso, Modelos de classificação, Regressão lógistica, *K-Nearest Neighbors*, Redes neurais, matriz de confusão, curva ROC.**

I. INTRODUÇÃO

Este trabalho investiga o desempenho de diferentes modelos de classificação na tarefa de prever o sucesso ou fracasso de uma aplicação para financiamento de pesquisa, com dados da Universidade de Melbourne [1]. Para o modelo linear, utilizamos a regressão logística, e para os modelos não lineares, utilizamos o *K-NN* e as redes neurais. Utilizamos a acurácia para avaliar o modelo e além dela, a matriz de confusão para avaliar melhor os resultados e entender melhor seu comportamento.

A principal contribuição deste estudo é oferecer uma visão abrangente sobre o desempenho de abordagens lineares e não lineares na predição de sucesso ou fracasso de aplicações de financiamento de pesquisa, buscando entender que modelos possuem mais acurácia e se predomina uma dinâmica linear ou não linear nessa predição.

II. MÉTODOS

A. Conjunto de dados

A base de dados histórica é composta por 8.707 propostas submetidas à Universidade de Melbourne nos anos de 2009 e 2010, contendo 1.882 preditores, mas devido à alta correlação entre certos preditores, foi utilizada a versão reduzida de 249 preditores. O sucesso ou fracasso da concessão do recurso é o que estamos buscando prever. A taxa de sucesso é de 48%, tendo assim uma distribuição bem balanceada, mas é interessante enfatizar que atualmente as taxas de sucesso de concessão são inferiores a 25%, dessa forma, é possível que os dados não descrevam tão bem a realidade atual. [1]

Os preditores incluíam diferentes tipos de medições e categorias, como ID do patrocinador, categoria da concessão, intervalo de valores de financiamento, área de pesquisa e departamento, sendo compostos por variáveis contínuas, contáveis e categóricas. [1]

Uma característica importante desse conjunto de dados é a alta proporção de valores ausentes, que representavam cerca de 83% dos preditores. Além disso, as amostras não eram independentes, pois o mesmo autor de proposta poderia aparecer várias vezes na base de dados. [1]

B. Pré-processamento dos dados

Para a construção dos modelos, os dados foram pré-processados. Os dados foram normalizados com relação à média e à variância para centralizá-los e distribuí-los de forma igual. E para a remoção de assimetria, foi utilizada a transformação de Yeo-Johnson.

A transformação de Yeo-Johnson possui comportamento semelhante ao método de Box-Cox, porém, funciona não só para conjuntos de dados com valores positivos, como para conjuntos com valores positivos e negativos. [2]

A transformação é dada pela fórmula (1).

$$\psi^{\mathrm{YJ}}(\lambda, \mathbf{x}) = \begin{cases} \frac{(\mathbf{x}+1)^{\lambda}-1}{\lambda} & \text{se } \lambda \neq 0,\ \mathbf{x} \geq 0, \\ \log(\mathbf{x}+1) & \text{se } \lambda = 0,\ \mathbf{x} \geq 0, \\ \frac{-[(-\mathbf{x}+1)^{2-\lambda}-1]}{2-\lambda} & \text{se } \lambda \neq 2,\ \mathbf{x} < 0, \\ -\log(-\mathbf{x}+1) & \text{se } \lambda = 2,\ \mathbf{x} < 0. \end{cases} \tag{1}$$

Em que x são os dados e λ é um parâmetro da transformação que define que tipo de resultado será obtido, se será uma

transformação linear, logarítmica ou potência dos dados. É possível estimar o λ através do estimador de máxima verossimilhança. [2]

C. Fundamentação Teórica

1) ***Regressão Logística****:* Modelo estatístico amplamente utilizado para problemas de classificação binária. Diferente da regressão linear, que prevê valores contínuos, a regressão logística aplica a função sigmoide para transformar uma combinação linear dos preditores em uma probabilidade entre 0 e 1. A equação principal do modelo é dada por (2):

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \sum \beta_i X_i)}} \tag{2}$$

onde β_0 é o termo independente e β_i são os coeficientes dos preditores X_i.

A interpretação da regressão logística baseia-se na relação entre os coeficientes β_i e as *odds* do evento de interesse. As *odds* são definidas como a razão entre as probabilidades de sucesso e de fracasso em (3):

$$\textit{odds} = \frac{P(Y = 1|X)}{1 - P(Y = 1|X)} \tag{3}$$

Tomando o logaritmo natural:

$$\log(\textit{odds}) = \beta_0 + \sum \beta_i X_i \tag{4}$$

A equação (4) mostra que a regressão logística modela a relação entre os preditores e a resposta como uma transformação logarítmica, permitindo interpretações diretas dos coeficientes: um valor positivo de β_i indica um aumento nas *odds* de $Y = 1$ para um aumento na variável X_i, enquanto um valor negativo indica o oposto.

O treinamento da regressão logística ocorre por meio da maximização da verossimilhança. A função de verossimilhança para um conjunto de dados de n observações é dada pela equação (5):

$$L(\beta) = \prod_{i=1}^{n} P(Y_i|X_i)^{Y_i}(1 - P(Y_i|X_i))^{1-Y_i} \tag{5}$$

O ajuste do modelo é realizado via métodos numéricos, como o algoritmo do gradiente descendente ou o método de Newton-Raphson, para encontrar os valores dos coeficientes β que maximizam a função de verossimilhança.

A regressão logística apresenta vantagens como o baixo custo computacional e ela fornece probabilidades preditivas, permitindo o ajuste de decisões. No entanto, ela assume que os preditores possuem uma relação linear com o logaritmo das *odds*, o que pode limitar seu desempenho em problemas altamente não lineares.

2) ***K-Nearest Neighbors (KNN)****:* O *K-Nearest Neighbors* (KNN) é um classificador baseado em instâncias que atribui a classe de um novo ponto considerando a maioria dos seus K vizinhos mais próximos. A similaridade entre pontos pode ser calculada por diferentes métricas de distância, tais como:

Distância Euclidiana: A distância Euclidiana é a métrica padrão em muitos problemas de aprendizado de máquina e é descrita por (6):

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{6}$$

onde x e y são vetores representando observações. Essa métrica mede a distância reta entre dois pontos em um espaço multidimensional.

Distância de Manhattan: A distância de Manhattan, utilizada no modelo otimizado, é calculada como:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{7}$$

Diferente da distância Euclidiana, a métrica de Manhattan considera apenas a soma das diferenças absolutas entre os atributos de dois pontos, tornando-a menos sensível a discrepâncias extremas. Essa característica pode ser vantajosa quando os dados contêm *outliers* ou quando as variáveis possuem escalas diferentes. Essa métrica é comumente usada quando os dados possuem uma estrutura de grade, como em mapas urbanos.

Distância de Minkowski: A distância de Minkowski é uma generalização das distâncias Euclidiana e de Manhattan e é definida por:

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{8}$$

O parâmetro p controla o comportamento da métrica:

- Se $p = 1$, a métrica equivale à distância de Manhattan.
- Se $p = 2$, a métrica equivale à distância Euclidiana.
- Para outros valores de p, a métrica pode ser ajustada para diferentes aplicações.

3) ***Redes Neurais****:* Modelos inspirados no funcionamento do cérebro humano e são especialmente úteis para modelar relações não lineares. Uma rede neural típica consiste em neurônios organizados em camadas (entrada, ocultas e saída), onde cada neurônio aplica uma função de ativação, como a ReLU (9) ou Sigmoide (10)

$$f(x) = \max(0, x) \quad \text{(ReLU)} \tag{9}$$

ou

$$f(x) = \frac{1}{1 + e^{-x}} \quad \text{(Sigmoid)} \tag{10}$$

As redes neurais são treinadas por meio do algoritmo de backpropagation, que ajusta os pesos utilizando métodos como gradiente descendente.

Uma rede neural pode ser composta por diversas camadas ocultas, onde cada camada contém múltiplos neurônios interconectados. A profundidade da rede influencia sua capacidade de aprendizado. Algumas variações populares incluem as Redes Neurais Convolucionais (CNNs) para processamento de imagens e as Redes Neurais Recorrentes (RNNs) para séries temporais.

O treinamento das redes neurais envolve a minimização de uma função de perda, como o erro quadrático médio ou a entropia cruzada, e o processo de otimização geralmente emprega variantes do gradiente descendente, como o Adam e RMSprop, para melhorar a convergência.

As redes neurais possuem alta capacidade de aprendizado e podem modelar padrões complexos. No entanto, elas exigem grande volume de dados para treinamento, possuem alto custo computacional e são menos interpretáveis em comparação a métodos mais simples, como a regressão logística. Além disso, o ajuste de hiperparâmetros, como a taxa de aprendizado e o número de camadas, pode ser um desafio.

*4) **Curvas ROC**:* As curvas *Receiver Operating Characteristic* (ROC) são utilizadas para avaliar o desempenho de modelos de classificação, especialmente em problemas binários, mostrando a capacidade do modelo de separar as classes corretamente. A curva ROC é obtida ao plotar a taxa de verdadeiros positivos (sensibilidade) contra a taxa de falsos positivos (1 - especificidade) para diferentes limiares de decisão.

Um modelo ideal teria uma curva ROC que se aproxima do canto superior esquerdo do gráfico, indicando alta sensibilidade e baixa taxa de falsos positivos. Já modelos que possuem uma linha diagonal na curva ROC indicam um desempenho equivalente ao de uma classificação aleatória. Além desses, modelos que possuem a curva ROC se aproximando do canto inferior direito do gráfico indicam alta sensibilidade e alta taxa de falsos positivos.

A área sob a curva ROC é o valor numérico que representa o desempenho global do modelo. A área varia de 0 a 1, sendo que:

- Área próxima de 1 indica um modelo altamente eficaz na separação das classes.
- Área = 0.5 indica um modelo que não tem poder discriminativo (classificação aleatória).
- Área próximo de 0 indica que o modelo está classificando as classes de forma invertida.

*5) **Matriz de Confusão**:* A matriz de confusão é uma ferramenta fundamental na avaliação de modelos de classificação, fornecendo uma visão detalhada do desempenho do classificador ao comparar as previsões feitas pelo modelo com os valores reais.

Para um problema de classificação binária, a matriz de confusão pode ser representada da seguinte forma:

Cada um desses valores representa um aspecto distinto do desempenho do modelo:

- **Verdadeiro Positivo (VP)**: Número de instâncias corretamente classificadas como positivas.

TABLE I
MATRIZ DE CONFUSÃO PARA CLASSIFICAÇÃO BINÁRIA

Real	**Valor Predito**	
	Sim	**Não**
Sim	Verdadeiro Positivo	Falso Negativo
Não	Falso Positivo	Verdadeiro Negativo

- **Falso Positivo (FP)**: Número de instâncias negativas incorretamente classificadas como positivas.
- **Falso Negativo (FN)**: Número de instâncias positivas incorretamente classificadas como negativas.
- **Verdadeiro Negativo (VN)**: Número de instâncias corretamente classificadas como negativas.

A matriz de confusão permite calcular diversas métricas para avaliar o desempenho do modelo:

- **Acurácia**: Mede a proporção de classificações corretas.

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + FN + VN} \tag{11}$$

- **Precisão**: Mede a proporção de predições positivas corretas.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{12}$$

- ***Recall* (Sensibilidade)**: Mede a proporção de exemplos positivos corretamente classificados.

$$\textit{Recall} = \frac{VP}{VP + FN} \tag{13}$$

- **Especificidade**: Mede a proporção de exemplos negativos corretamente classificados.

$$\text{Especificidade} = \frac{VN}{VN + FP} \tag{14}$$

- **F1-Score**: Média harmônica entre precisão (12) e *recall* (13), sendo útil para balancear os dois aspectos.

$$F1 = 2 \times \frac{\text{Precisão} \times \textit{Recall}}{\text{Precisão} + \textit{Recall}} \tag{15}$$

A matriz de confusão é amplamente utilizada na avaliação de classificadores em diversas áreas, como reconhecimento de padrões, diagnóstico médico e detecção de fraudes. Ela permite compreender melhor os erros do modelo e direcionar melhorias no ajuste de hiperparâmetros e seleção de features.

Para problemas de classificação com mais de duas classes, a matriz de confusão é expandida para um formato de $n \times n$, onde n é o número de classes, onde cada célula (i, j) da matriz representa o número de instâncias da classe real i que foram preditas como classe j.

III. Resultados

A. Regressão Logística

Com a Regressão Logística foram obtidos os seguintes resultados:

Métricas de desempenho:

- Acurácia: 0.86
- Recall: 0.82
- Especificidade: 0.89

- Precisão: 0.81
- F1 Score: 0.81

Curva ROC e Matriz de Confusão:
Obtivemos a curva ROC para a Regressão Logística:

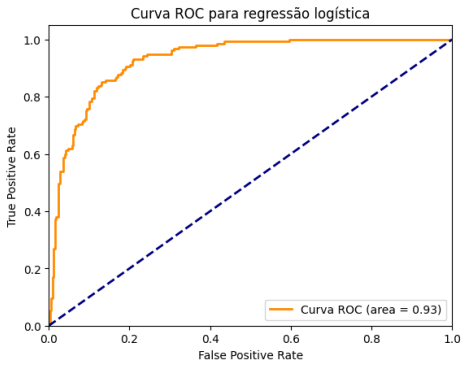

Fig. 1. Curva ROC para regressão logística

E a matriz de confusão:

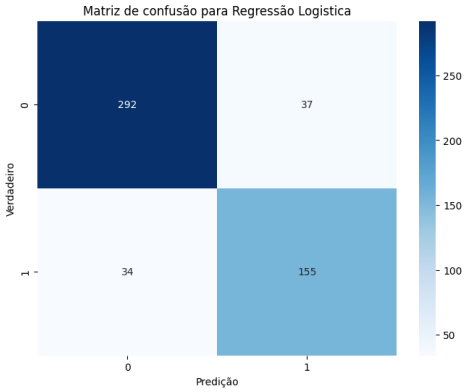

Fig. 2. Matriz de confusão para Regressão Logística

Área sob a Curva ROC:
A Regressão Logística obteve uma área sob a curva ROC de 0.93, o que indica um excelente desempenho na separação entre as classes. Comparando com outros modelos, a regressão logística superou o KNN, que teve uma área de 0.85, e demonstrou um nível de distinção entre classes próximo ao da Rede Neural, que teve uma área de 0.92.

Os resultados da regressão logística mostram um excelente equilíbrio entre precisão e recall, evidenciado pelo F1-score de 0.81. O alto valor de especificidade (0.89) também sugere que o modelo é altamente eficiente na detecção correta das instâncias negativas, minimizando falsos positivos e falsos negativos.

B. K-Nearest Neighbors

Para a escolha dos melhores hiperparâmetros do *K-Nearest Neighbors*, foi utilizado o método de *Grid Search*, testando diferentes combinações de parâmetros. Os seguintes valores foram explorados:

- **Número de vizinhos (`n_neighbors`):** 15, 17, 23, 25, 27, 29
- **Pesos (`weights`):** uniform, distance
- **Métrica de distância (`metric`):** euclidean, manhattan, minkowski

Após a busca, os melhores parâmetros encontrados foram:

- **Melhores parâmetros:**
  - `n_neighbors`: 25
  - `weights`: distance
  - `metric`: manhattan
- **Melhor acurácia na validação:** 0.7934

Os resultados obtidos com o modelo otimizado foram:
Métricas de desempenho:

- Acurácia: 0.79
- Recall: 0.69
- Especificidade: 0.85
- Precisão: 0.73
- F1 Score: 0.71

Análise dos Resultados:
O desempenho do KNN foi moderado, com uma acurácia de 79%. No entanto, o *recall* de 0.69 indica que o modelo pode estar deixando passar muitas instâncias positivas, ou seja, uma taxa relativamente alta de falsos negativos. A precisão de 0.73 sugere que o modelo tem um equilíbrio razoável entre identificar corretamente casos positivos e evitar falsos positivos.

A escolha de $k = 25$ e o uso da métrica de distância *manhattan*, com pesos baseados na distância, ajudaram a otimizar o modelo. No entanto, a sensibilidade do KNN à escolha dos hiperparâmetros pode ter impactado sua generalização.

Curva ROC e Matriz de Confusão:
Obtivemos a curva ROC para o *K-Nearest Neighbors*:

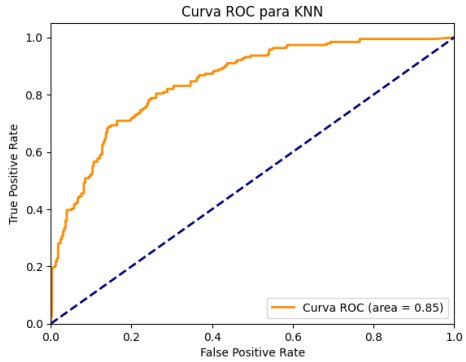

Fig. 3. Curva ROC para o *K-Nearest Neighbors*

E a matriz de confusão:

Área sob a Curva ROC:
A área sob a curva ROC obtida foi de 0.85, indicando um bom desempenho na separação entre classes. Embora o modelo apresente uma boa capacidade preditiva, seu desempenho inferior ao da rede neural e a sensibilidade a variações nos dados podem torná-lo uma escolha menos robusta para este problema.

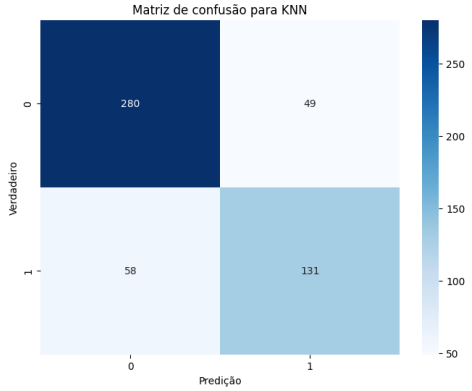

Fig. 4. Matriz de confusão para o *K-Nearest Neighbors*

C. Redes Neurais

Arquitetura da Rede Neural:

A rede neural utilizada é composta por cinco camadas densas com a seguinte configuração:

Camada	Número de Neurônios	Ativação
Entrada	$\lvert X_{\text{train}} \rvert$	-
Oculta 1	512	ReLU
Oculta 2	256	ReLU
Oculta 3	128	ReLU
Oculta 4	64	ReLU
Saída	1	Sigmoid

TABLE II
ESTRUTURA DA REDE NEURAL

Além disso, foram utilizadas as seguintes técnicas de regularização e otimização:

- **Dropout:** 0.5 aplicado após cada camada oculta.
- **Batch Normalization:** antes da ativação em cada camada oculta.
- **Ativação da saída:** Sigmoid.
- **Loss Function:** Binary Cross-Entropy (BCE).
- **Otimização:** Adam com taxa de aprendizado de 10^{-4}.
- **Treinamento:** 50 épocas com *batch size* de 32.
- **Early Stopping:** monitorando a *val_loss* com paciência de 10 épocas e restauração dos melhores pesos.

Análise dos Resultados:

Com as Redes Neurais foram obtidos os seguintes resultados:

Métricas de desempenho:

- Acurácia: 0.84
- Recall: 0.83
- Especificidade: 0.84
- Precisão: 0.75
- F1 Score: 0.79

O uso de *dropout* e *batch normalization* ajudou a evitar o *overfit*, permitindo uma melhor generalização para os dados de teste.

Curva ROC e Matriz de Confusão:

Obtivemos a curva ROC para as Redes Neurais:

E a matriz de confusão:

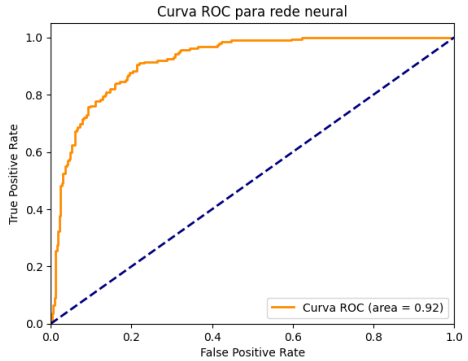

Fig. 5. Curva ROC para as Redes Neurais

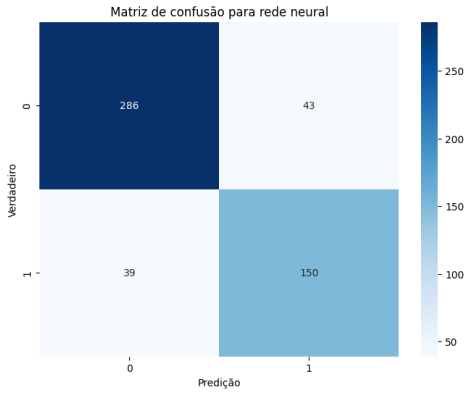

Fig. 6. Matriz de confusão para as Redes Neurais

Área sob a Curva ROC:

A rede neural obteve uma AUC elevada, indicando uma boa separação entre as classes positivas e negativas. Isso sugere que o modelo foi capaz de aprender padrões complexos nos dados, equilibrando corretamente precisão e recall.

REFERENCES

[1] Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. doi:10.1007/978-1-4614-6849-3

[2] Yeo, I.-K., & Johnson, R. A. (2000). *A new family of power transformations to improve normality or symmetry*. Biometrika, 87(4), 954–959. doi:10.1093/biomet/87.4.954

[3] Pearson, K. (1895). Notes on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242. doi:10.1098/rspl.1895.0041.

[4] Karhunen, K. (1947). Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Annales Academiae Scientiarum Fennicae. Series A. I. Mathematica-Physica*, vol. 37, pp. 1–79.

[5] Loève, M. (1948). Fonctions aléatoires de second ordre. *Revue Scientifique*, vol. 86, pp. 195–206.

Contribuição	
Conceitualização	Mateus
Metodologia	Mateus
Software	Kelvin
Validação	Gabriel
Análise formal	Pedro
Redação	Gabriel, Kelvin, Mateus, Pedro